Andrew’s Leap 2003
Introductory Programming

Project list, July 7

As we near the end of programming basics, we will begin to work on some larger “projects” that will require increased time, creative thought, and brainpower. Many of the projects below find inspiration from Dr. Rudich’s morning lectures and similar mathematical topics. Others are extensions of basic topics that we have learned in the programming section. You may find it helpful to work in groups of two to three to tackle these problems, though you may work alone if you like. There is not necessarily a “right” answer to any one problem.

Some of these “projects” are complete projects, and others you will be able to incorporate into larger projects and investigations.

You will not be graded on any of these problems, and you may do as few or as many as you like.

If you have an idea for a similar project, tell me so we may look for a way to solve it. As always, ask me any questions you have about the project, though I may leave it to you to find any appropriate algorithm to solve your problem. In fact, I suggest that you map out on paper an algorithm to solve the problem before you dive into coding.

1. Write a program that allows a user to simulate the flipping of a pancake stack of size 5 (or of size *n* if you wish). The user should input, as an array, the starting configuration of the stack, and, as an integer, the number of pancakes to be flipped. The program should then output the flipped stack of pancakes. For example, if the user inputs the numbers 3,2,4,5,1 as the pancake stack and 3 as the number of pancakes to be flipped, the program should output the numbers 4,2,3,5,1 as the flipped stack. The part of the program that actually does the flipping should be contained in its own method. This method should take two inputs: an `int[]` (the stack itself) and an `int` (the number of pancakes to be flipped), and should return an `int[]` (the new stack).

2. Write a program that prints a grid of dimensions *x* by *y*. The user should input the number *x* (width) and *y* (height), and the program should print a grid of those dimensions using the symbols +, −, and |. For example, a 5 by 2 grid should be displayed as follows:

   ```
   +-+-+-+-+-+
   | | | | | |
   +-+-+-+-+-+
   | | | | | |
   +-+-+-+-+-+
   ```

   You’ll need to use nested FOR loops, and you’ll need to know the following print commands:

   `Std.out.println()` always begins a new line after the output.
   `Std.out.print()` does *not* begin a new line after the output.

3. Construct a calculator. Your user should input one number (think about the input type), then an operator (+, −, *, /, or ^), then a second number. The calculator should perform the operation on the two numbers. You will need various IF statements to check to see if the input is legal and to perform a different operation based on the input of the operator. Place the code for the various operations in different methods. (Challenge: bearing in mind the lessons of Prof. Bryant's lecture, check to ensure that there has not been a data overflow that will cause an incorrect answer.)

Below are a few more tricks that may be necessary to use in your code:

Inputting a character
I have found that the `Std.in.readChar()` method does not work properly. Therefore, it is necessary to use the `Std.in.readWord()` method instead and to take the first (and, in this case, only) character from that word. This can be done as follows:

```
char op;
String op1;
op1 = new String(Std.in.readWord());
op = op1.charAt(0);
```

Using logical symbols
Within the test clause of a loop (the part in the beginning in parentheses), it is sometimes necessary to use a variety of symbols, sometimes in combination, to generate the desired logical statement. The following is a list of logical symbols:

`==`	equals
`<+`	is less than or equal to
`>=`	is greater than or equal to
`!+`	is not equal to
`&&`	and
`||`	or

For example, if you want to see if a number is greater than 2 but not equal to ten, you could begin your IF statement with the following line:

```
if ( number > 2 && number != 10 ) { … }
```